



Low-Level Advanced Design of True Random Number Generators Based on Truly Chaotic Digital Nonlinear Oscillators in FPGAs

Tommaso Addabbo, Ada Fort, Riccardo Moretti^(✉), Marco Mugnaini, and Valerio Vignoli

Department of Information Engineering and Mathematics, University of Siena,
Via Roma 56, 53100 Siena, (SI), Italy
{addabbo,ada,moretti,mugnaini,vignoli}@diism.unisi.it
<http://www.diism.unisi.it/>

Abstract. Recently, a new class of circuits named Digital Nonlinear Oscillators (DNOs) has been proposed for the design of fully digital True Random Number Generators (TRNGs). In this work we discuss the low-level advanced design of TRNGs based on chaotic DNOs, specialized for FPGAs. In detail, starting from a specific DNO topology, we discuss technical solutions to implement these systems exploiting FPGA device primitives. The proposed solutions have been characterized by means of exhaustive measurement campaigns to assess and investigate the impact on the entropy of both chip-to-chip and intra-device variability.

1 Introduction

In information security, True Random Number Generators (TRNGs) are circuits designed to generate truly random binary sequences to be used in cryptographic protocols. These circuits implement entropy sources, being able to generate unpredictable random bits *deemed to be sufficiently secure* for the considered application [1, 2].

A relevant effort has been made by researchers to investigate solutions suitable for being implemented in digital hardware. Traditionally, such sources of entropy, designed in silicon integrated circuits, are based on the exploitation of electronic noise and meta-stable cells, combined in different ways [3–17]. TRNGs must be distinguished from Pseudo Random Number Generators (PRNGs), that are deterministic digital circuits aiming to simulate a truly random source, i.e., not capable to generate information by themselves [1, 2, 18].

Recently, a new class of circuits named Digital Nonlinear Oscillators (DNOs) has been proposed for the design of fully digital TRNGs [19–22]. As defined in [19–22], DNOs are networks of electronic digital circuits, each one originally designed to behave as an asynchronous logic gate, implementing autonomous nonlinear dynamical systems, exhibiting oscillations in the time-continuous

domain. In [19] it has been shown that DNOs can define dynamical systems *supporting structurally stable chaotic dynamics*.

In this work we discuss the low-level advanced design of TRNGs based on chaotic DNOs, specialized for FPGAs.

2 DNOs as Dynamical Systems

Conceptually, the topology of a DNO can be represented as the interconnection of subcircuits called Elementary Logic Blocks (ELBs).

In Fig. 1a a DNO made of six ELBs is presented. The asynchronous domain, in which the DNO operates as the entropy source, has been separated by the surrounding synchronous domain, in which a digital state machine can be used to implement a TRNG. The two domains are joined by low-complexity synchronous circuitry that, in the simplest case, performs the uniform 1-bit sampling of the DNO *analog* dynamics, e.g., by means of a single D flip flop. The interface between the synchronous and the asynchronous domains represents another possible source of randomness, since the flip-flop can be affected by metastability. This condition represents an advantage for the circuit purposes, because it increases the entropy provided by the source.

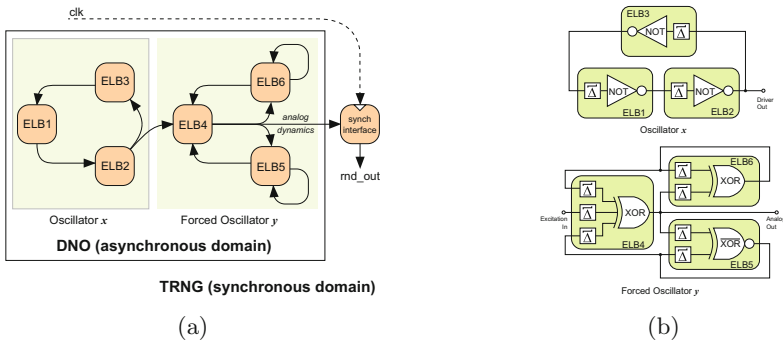


Fig. 1. (a) A Digital Nonlinear Oscillator (DNO) made of six Elementary Logic Blocks (ELBs). The synchronous and asynchronous domains are joined by low-complexity interfacing circuitry (a single D flip flop in the simplest case). (b) DNO based on the topology shown in Fig. 1a suitable for being implemented in FPGAs. Each ELB incorporates a logic functionality and the active digital routing of signals, reported at their inputs.

The specific internal structure of the DNO reported in the figure can be interpreted as a coupled oscillators. A special case of coupled oscillators is obtained when an autonomous dynamical system \mathbf{x} is used to generate a driving signal exciting a second dynamical system \mathbf{y} :

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \\ \dot{\mathbf{y}} = \mathbf{g}(\mathbf{x}, \mathbf{y}), \end{cases} \quad (1)$$

being $\mathbf{x} : \mathbb{R} \rightarrow \mathbb{R}^N, \mathbf{y} : \mathbb{R} \rightarrow \mathbb{R}^M$ real-valued functions of time t , and \mathbf{f}, \mathbf{g} nonlinear smooth real-valued functions of \mathbf{x} and \mathbf{y} , respectively. If $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ and $\dot{\mathbf{y}} = \mathbf{g}(\mathbf{0}, \mathbf{y})$ define two periodic dynamical systems, we may call \mathbf{y} in (1) the *forced oscillator*, being \mathbf{x} the *forcing periodic driver*.

As it can be seen in Fig. 1b, the driving oscillator is composed of three ELBs, and is topologically equivalent to a Ring Oscillator made of three inverting gates. On the other hand, the forced oscillator \mathbf{y} is composed by three ELBs implementing *xor* or *nxor* boolean gates. As represented in the figure, each ELB incorporates both a logic functionality and the active digital routing of its input signals, represented in the scheme as active delays. In the next Section, the low-level FPGA implementation of the DNO is discussed in detail.

3 FPGA Hardware Implementation

The low-level design of the circuit shown in Fig. 1b in a FPGA, discussed hereafter, is a constrained implementation, in which we took control of the hardware resources utilization, using device primitives. In this section we discuss how this constrained implementation should be performed. For lack of space, the HDL description of the investigated system implemented referring to a Xilinx Artix 7 FPGA is reported in [23].

The ELBs low-level design have been carried out using device primitives. The access to these resources is guaranteed invoking the UNISIM library by Xilinx, in any entity directly involved in the design [23]. The logic functionality of any ELB can be implemented by means of one LUT: this means that the DNO in Fig. 1b requires a total of six LUTs (less than the resources available in two Artix 7 slices). For clarity of presentation, each ELB in [23] has been associated to a VHDL entity. In the code, special directives have to be used to force the resource utilization at specific chip locations. Since the DNO is an asynchronous circuit, registers have not to be used. The active routing, necessary to interconnect the ELBs, completes the DNO architecture.

In DNO design, *combinatorial loops are mandatory*. Normally, combinatorial loops should be avoided in digital designs. They occur when a combinatorial logic feed back to itself without registers, potentially creating logic race conditions or spoiling the timing analysis during the design phases. For these reasons, most design tools generate Design Rules Check (DRC) errors during the synthesis. To allow combinatorial loops, specific directives to enable the synthesis of intentional loops in asynchronous digital structures have been provided [23].

To its minimal terms, the synchronization interface in Fig. 1a can be reduced to a single D flip-flop performing, at the same time, the 1-bit A/D conversion and the uniform sampling of the chaotic signal (provided by the ELB4 in Fig. 1a). To optimize hardware utilization, the 1-bit register FF primitive has been located in the same slice of the 6LUT primitive implementing the logic functionality of the ELB4. The Synchronization Interface takes part in the static timing analysis of the full design, and its constrained location may have an impact on the successful meet of timing constraints. In complex projects, this issue has to be carefully addressed by the designer.

Finally, in a FPGA the configurable routing is organized by means of programmable switches and connection boxes, according to a hierarchical architecture offering local and regional connectivity. Once that the ELBs have been placed in specific chip locations, the final routing is left to the compiler. To minimize the impact of the DNO on the general project routing, it is recommended to have the ELBs concentrated in few slices, placed close to each others. As discussed in the next Section, this also mitigates the impact of hardware variability on the entropy levels of the TRNG.

4 Impacts of Variability: Experiments

The entropy that can be extracted from a chaotic system is always sensitive to the perturbation of its dynamical parameters. The magnitude of the perturbation is linked to its effects on the entropy in a nonlinear way and, with the exception of few cases, the issue has to be investigated by means of numerical simulations or experiments.

In this Section, we present an exhaustive measurement campaign based on experiments devised to assess the impact of both the chip-to-chip variability and the intra-device variability on the entropy. To this aim, we designed 16 instances of the DNO, in different chip areas of the FPGA, repeating the measurements for six different chips (using the same slice locations), reaching a total of 96 DNO instances. The measurements have been performed implementing the system in Xilinx Artix 7 xc7a35 FPGAs, using an internal sampling clock frequency of 400 MHz.

For each case we estimated the Average Shannon Redundancy (ASR) evaluated on binary words of 10 bits, defined as $ASR_{10} = 1 + \frac{1}{10} \sum_{i=1}^{2^{10}} P(w_i) \log_2 P(w_i)$, [bit/sym], where $P(w_i)$ is the generation probability for $w_i \in \{0, 1\}^{10}$, being the summation extended to the 1024 possible binary words of 10 bits. The estimations were obtained acquiring streams of 1 million bits per experiment at room temperature.

4.1 Condensed Layout

Logic resources in a Xilinx Artix 7 FPGA are organized as a matrix of Configurable Logic Blocks (CLBs), each one containing two slices, and each slice being composed of four 6-input Look Up Tables (LUTs) and eight storage elements.

In the upper subplots of Fig. 2 we reported the experimental results highlighting the effects on the ASR of both the chip-to-chip variability and the intra-device variability, for a condensed layout in which the entire DNO in Fig. 1a was concentrated in two slices (same CLB), including the sampling flip flop D. The percentile levels L_x , expressed in bit/sym for $x = 10, 50, 80, 90, 95$, were estimated on the base of the entire data set (96 DNO instances). Red square symbols were used to highlight the chip location 1 (A) or the chip number 1 (B).

As it can be appreciated from the figure, 90% of the chaotic DNOs are capable to provide outstanding levels of ASR_{10} below $L_{90} = 0.077$ bit/sym with a

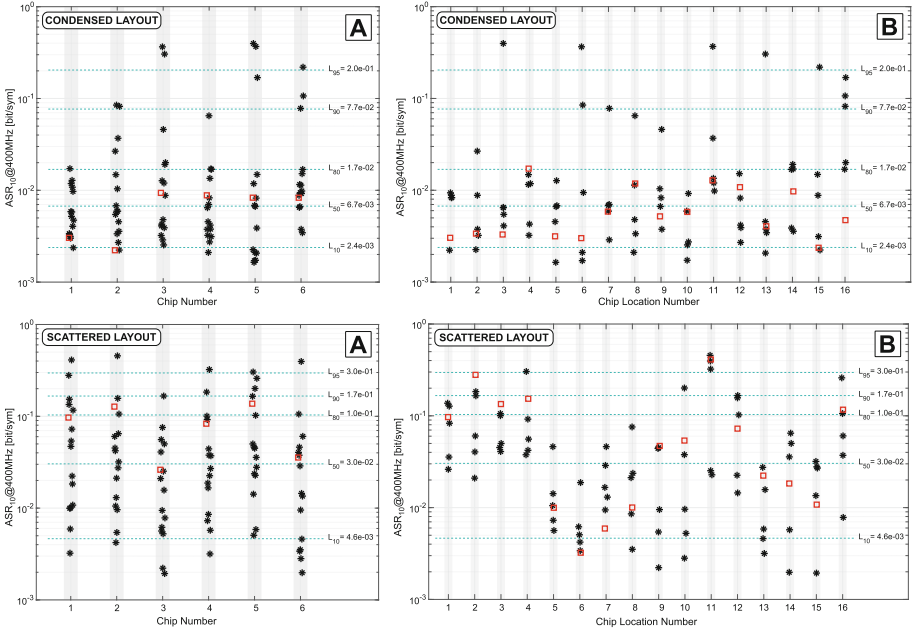


Fig. 2. Effects on the ASR of chip-to-chip and intra-device variabilities, for a condensed and a scattered layout. Red square symbols were used to highlight the chip location 1 (A) or the chip number 1 (B). (Color figure online)

sampling frequency of 400 MHz. This corresponds, in principle, to 369.2 Mbit/s of truly random information generated with the minimal usage of only two FPGA slices.

4.2 Scattered Layout

The effects on the ASR of both the chip-to-chip variability and the intra-device variability have been evaluated as a function of the routing complexity. As confirmed by experimental results, adopting a scattered layout in which the ELBs are separated by longer distances in the CLBs matrix increases the impact of variability. The same experiments presented in the previous Subsection have been repeated adopting a scattered layout in which each LUT of the DNO is positioned in a different FPGA CLB.

When adopting a scattered layout, the ELBs are connected through an higher number of switch boxes. As it can be appreciated from the lower subplots of Fig. 2, this solution worsen the entropy, in average, enhancing the effects of variability. However, it is worth noting that the best cases in both of the layout (10th percentile) share similar levels of ASR.

In this case 90% of the chaotic DNOs are capable to provide levels of ASR₁₀ below $L_{90} = 0.17$ bit/sym with a sampling frequency of 400MHz. This corre-

sponds, in principle, to 332.0 Mbit/s of information. The results are still exceptional, considering the reduced amount of resource utilization.

5 Conclusion

In this work we have discussed the low-level advanced design of TRNGs based on chaotic DNOs, specialized for FPGAs. In detail, starting from a specific DNO topology, we have discussed technical solutions to implement these systems exploiting FPGA device primitives, performing constrained layout implementations. The proposed solutions, capable to provide high entropy levels at a minimal cost of resources utilization, have been characterized by means of exhaustive measurement campaigns to assess and investigate the impact on the entropy of both chip-to-chip and intra-device variability.

According to the observed results, compact layouts allow the system to achieve higher performances in terms of generated entropy. This implies that we can variate the circuit performance simply acting on the routing. An interesting aspect to be investigated would be the effect of the chosen layout on the resulting system power consumption. This kind of investigation will be realized in the future works, as we plan to adapt the presented low-level design approaches for the design of fully digital ASIC based TRNGs.

References

1. Acosta, A., Addabbo, T., Tena-Sanchez, E.: Embedded electronic circuits for cryptography, hardware security and true random number generation: an overview. *Int. J. Circuit Theory Appl.* **45**(2), 145–169 (2017)
2. NIST Special Publication 800-22 Rev. 1a: A statistical test suite for random and pseudorandom number generators for cryptographic applications, April 2010. <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf>
3. Kwok, S.H.M., Lam, E.Y.: FPGA-based high-speed true random number generator for cryptographic applications. In: *TENCON 2006– IEEE Region 10 Conference*, pp. 1–4 (2006)
4. Öztürk, H.S., Ergün, S.: A digital random number generator based on chaotic sampling of regular waveform. In: *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 178–181 (2020)
5. Demir, K., Ergun, S.: Random number generators based on irregular sampling and Fibonacci-Galois ring oscillators. *IEEE Trans. Circuits Syst. II Express Briefs* **66**(10), 1718–1722 (2019)
6. Anandakumar, N.N., Sanadhya, S.K., Hashmi, M.S.: FPGA-based true random number generation using programmable delays in oscillator-rings. *IEEE Trans. Circuits Syst. II Express Briefs* **67**(3), 570–574 (2019)
7. Carreira, L.B., Danielson, P., Rahimi, A.A., Luppe, M., Gupta, S.: Low-latency reconfigurable entropy digital true random number generator with bias detection and correction. *IEEE Trans. Circuits Syst. I Regul. Pap.* **67**(5), 1562–1575 (2020)
8. Sivaraman, R., Sridevi, A., Rajagopalan, S., Janakiraman, S., Rengarajan, A.: Design and analysis of ring oscillator influenced beat frequency detection for true random number generation on FPGA. In: *2019 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–6, January 2019

9. Tao, S., Yu, Y., Dubrova, E.: FPGA based true random number generators using non-linear feedback ring oscillators. In: 2018 16th IEEE International New Circuits and Systems Conference (NEWCAS), pp. 213–216, June 2018
10. Sui, C., Bai, S., Zhu, T., Cheng, C., Beetner, D.: New methods to characterize deterministic jitter and crosstalk-induced jitter from measurements. *IEEE Trans. Electromagn. Compat.* **57**(4), 877–884 (2015)
11. Raitza, M., Vogt, M., Hochberger, C., Pionteck, T.: Raw 2014: random number generators on FPGAs. *ACM Trans. Reconfigurable Technol. Syst.* **9**(2), 15:1–15:21 (2015)
12. Golic, J.D.J.: New methods for digital generation and postprocessing of random data. *IEEE Trans. Comput.* **55**(10), 1217–1229 (2006)
13. Wang, X., et al.: High-throughput portable true random number generator based on Jitter-Latch structure. *IEEE Trans. Circuits Syst. I Regul. Pap.* **68**(2), 741–750 (2021)
14. Tsoi, K.H., Leung, K.H., Leong, P.H.W.: Compact FPGA-based true and pseudo random number generators. In: 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM 2003, April 2003, pp. 51–61 (2003)
15. Hata, H., Ichikawa, S.: FPGA implementation of metastability-based true random number generator. *IEICE Trans. Inf. Syst.* **E95.D**(2), 426–436 (2012)
16. Wiczorek, P.Z.: An FPGA implementation of the resolve time-based true random number generator with quality control. *IEEE Trans. Circuits Syst. I Regul. Pap.* **61**(12), 3450–3459 (2014)
17. Wu, X., Li, S.: A new digital true random number generator based on delay chain feedback loop. In: IEEE International Symposium on Circuits and Systems (ISCAS), vol. 2017, pp. 1–4 (2017)
18. Addabbo, T., Alioto, M., Fort, A., Rocchi, S., Vignoli, V.: Low-hardware complexity PRBGs based on a piecewise-linear chaotic map. *IEEE Trans. Circuits Syst. II Express Briefs* **53**(5), 329–333 (2006)
19. Addabbo, T., Fort, A., Moretti, R., Mugnaini, M., Takaloo, H., Vignoli, V.: A new class of digital circuits for the design of entropy sources in programmable logic. *IEEE Trans. Circuits Syst. I Regul. Pap.* **67**(7), 2419–2430 (2020)
20. Addabbo, T., Fort, A., Moretti, R., Mugnaini, M., Vignoli, V., Garcia-Bosque, M.: Lightweight true random bit generators in PLDs: figures of merit and performance comparison. In: 2019 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5, May 2019
21. Addabbo, T., Fort, A., Mugnaini, M., Vignoli, V., Garcia-Bosque, M.: Digital non-linear oscillators in PLDs: Pitfalls and open perspectives for a novel class of true random number generators. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5, May 2018
22. Addabbo, T., Fort, A., Moretti, R., Mugnaini, M., Vignoli, V.: Analysis of a circuit primitive for the reliable design of digital nonlinear oscillators. In: 2019 15th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME), pp. 189–192, July 2019
23. Addabbo, T., Fort, A., Moretti, R., Mugnaini, M., Vignoli, V.: DNO Xilinx Artix 7 hardware implementation. <http://www3.diism.unisi.it/~addabbo/ApplePies2021/HDL.html>